

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**

**THIS PAGE BLANK (USPTO)**

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-134693  
 (43)Date of publication of application : 23.05.1995

E5887

(51)Int.Cl.

G06F 15/00  
 G06F 9/46

(21)Application number : 05-280824  
 (22)Date of filing : 10.11.1993

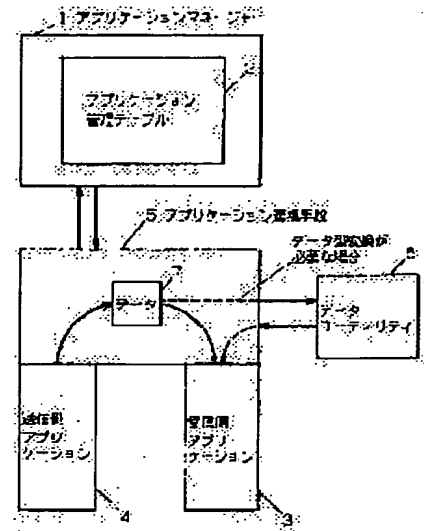
(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD  
 (72)Inventor : SEKINE SATOSHI  
 HATA TSUTOMU

## (54) APPLICATION COOPERATION DEVICE

## (57)Abstract

PURPOSE: To provide the flexibility of a data type switched among applications and application cooperation for a type requesting only the minimum handling time to an end user when data is switched.

CONSTITUTION: An application manager 1 is provided with an application table 2 in the inside, and processes access to the table by an application cooperation means 5. The table is provided with a name recognizable by the user of the application and an identifier identifiable by a computer system as elements, and it is used in the switching of cooperation partners opponent from the name inputted from the user to the identifier when the data is processed actually. A data utility 6 is automatically called when type switching is requested at the application cooperation means, and performs type witching.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(43)公開日 平成7年(1995)5月23日

### 技術表示箇所

審査請求 未請求 請求項の数2 OL (全 5 頁)

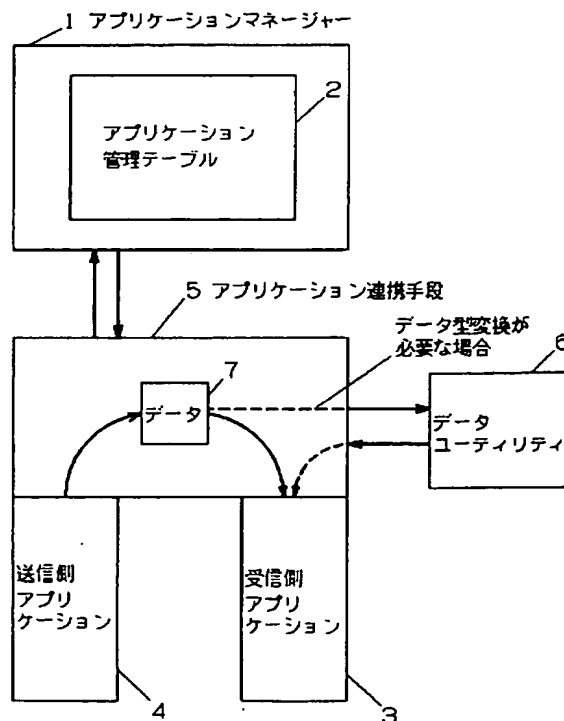
(74) 代理人 弁理士 小鍬治 明 (外2名)

(54) 【発明の名称】      アプリケーション連携装置

(57) 【要約】

【目的】 アプリケーション間で交換されるデータの型の柔軟性と、データ交換の際にはエンドユーザーに最小の手間だけしか求めない形のアプリケーション連携を提供することを目的とする。

【構成】 アプリケーションマネージャー 1 はアプリケーションテーブル 2 を内部に持ち、アプリケーション連携手段 5 によるテーブルへのアクセスを処理する。テーブルはアプリケーションのユーザーが認識できる名前と計算機システムが識別できる識別子を要素として持ち、連携相手をユーザーが入力した名前から実際に処理する際の識別子への変換に利用される。データユーティリティ 6 は、アプリケーション連携手段において型変換が必要となった時に自動的に呼び出され型変換を行う。



## 【特許請求の範囲】

【請求項1】 アプリケーションを管理するアプリケーションマネージャーと、アプリケーションがアプリケーション連携のために使用するアプリケーション連携手段とを備え、前記アプリケーションマネージャーがエンドユーザーがアプリケーションを認識するためのアプリケーション名と計算機システムがアプリケーションを識別するための識別子からなるアプリケーション管理テーブルを有し、連結を要求するアプリケーションから指定された連結先のアプリケーション名に対応する識別子を返し、前記アプリケーション連携手段が前記識別子で特定されるアプリケーションと連結を要求するアプリケーションとの連結を行うアプリケーション連携装置。

【請求項2】 アプリケーション間で交換されるデータの型変換を行うデータユーティリティを備え、データ連結手段がアプリケーションから指定されたデータ型に基づき前記データユーティリティにデータ変換を指示することを特徴とする請求項1記載のアプリケーション連携装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、計算機システム上で複数のアプリケーション間のデータの交換に関わるアプリケーション連携装置に関するものである。

## 【0002】

【従来の技術】 近年、計算機システムの発展において、複数のアプリケーションを同時に動作させることができるようになった。また、エンドユーザーに対する計算機インターフェースの発展に伴って、ディスプレイ上に複数のウィンドと呼ばれる、端末機能やアプリケーション機能を同時に表示し、あたかもユーザーが複数のアプリケーションを並列に扱っているような状況をつくり出せるようになってきた。

【0003】 また、いろいろな種類のアプリケーションが開発され、それらのアプリケーションも多くの種類のデータを扱えるようになり、時には別々のアプリケーションが同じ型のデータを扱うこともある。

【0004】 このような状況において、1つのディスプレイ上に同時に存在する複数のアプリケーション管でのデータ交換によるアプリケーション連携の要求が高まっている。

【0005】 その1つの実現方法として一般に浸透しているのは、ディスプレイに表示されているデータの一部を切り取り、他のアプリケーションの指定された部分に張り付けるマウス等によるカットアンドペースト方式やファイルという単位でのデータをデータ交換尾対象としたドラッグアンドドロップ方式である。

【0006】 第3図はカットアンドペーストの具体的実現例、第4図においてドラッグアンドドロップの具体的実現例を示す。

【0007】 まず、第3図においては、31はマウスによって選択された領域を示し、32はマウスの移動、33は選択された領域が張り付けられた部分を示す。この手法では、あるアプリケーションにおいてディスプレイに表示されている部分のデータを他のアプリケーションにおいてディスプレイに表示されている指定した場所に転送することができる。

【0008】 また、第4図においては、41は複数のファイルを含むホルダーの表示ウィンド、42はファイルを1つの処理単位として動作する目的アプリケーション、43は目的アプリケーションに転送しようとしているファイル、44はファイルを掴んだ（ドラッグ）後、目的アプリケーションを示すディスプレイの目印の上に落そう（ドロップ）としてファイルを移動させている状態を示す。この手法では、ファイルを1つの操作単位としたデータ交換が実現できる。

【0009】 別の実現方法として、ウィンド間連携におけるデータ連携方式（特開昭62-243030号）がある。この手法では、あるアプリケーションで生成されたデータをウィンド管理部における会話的処理を利用して別のアプリケーションのデータとして利用するデータ連携方式である。

## 【0010】

【発明が解決しようとする課題】 しかし、上記の方法ではいくつかの問題があった。まず、カットアンドペーストの方式では、切り取りたい領域も、張り付けたい位置もディスプレイ上で表示されていなければいけないという点がある。また交換するデータの型は完全に一致していなければならず、実際には単なる文字配列のみが扱える対象である場合がほとんどである。

【0011】 次に、ドラッグアンドドロップ方式では、処理単位がファイルであるため、アプリケーションはファイルという単位を用いて出来ているものではなくならず別の形態での使用を目的とする場合には利用できない。また、データを全体に対する部分として扱うこともできず柔軟性がない。

【0012】 さらに、特開昭62-243030号のウィンド間連携におけるデータ連携方式では、データ交換においては会話的手段を用いて、対象アプリケーションやデータ型をエンドユーザーが会話処理において明示的に指定しなければいけなかったため、エンドユーザーにとって操作が繁雑であるという問題があった。

【0013】 本発明は上記従来の課題を解決するもので、アプリケーション間で交換されるデータの型の柔軟性と、データ交換の際にはエンドユーザーに最小の手間だけしか求めない形のアプリケーション連携装置を提供することを目的とする。

## 【0014】

【課題を解決するための手段】 上記課題を解決するために、本発明のアプリケーション連携装置は、アプリケー

ションを管理するアプリケーションマネージャーと、アプリケーションがアプリケーション連携のために使用するアプリケーション連携手段と、アプリケーション間で交換されるデータのデータ型についての処理を扱う手段であるデータユーティリティとを備え、アプリケーションマネージャーの要素としてエンドユーザーが認識できるアプリケーションを示すアプリケーションの名前と計算機システムがデータの交換を行なうために必要なアプリケーションの識別子からなるアプリケーション管理テーブルを有する構成となっている。

#### 【0015】

【作用】この構成によって、計算機システム上で複数のアプリケーション間のデータの交換に関わるアプリケーション連携処理において、アプリケーションがアプリケーション連携のために使用するアプリケーション連携手段によって希望するデータ型を指定でき、交換されるデータの型が希望するデータ型と異なる場合は、データユーティリティによって自動的にデータ型の変換が行なわれ、エンドユーザーがデータ型を指定しななければならない煩わしさから解放される。

【0016】また、アプリケーション管理テーブルを有するアプリケーションマネージャーによりデータ交換希望相手のアプリケーションをエンドユーザーが認識できる名前によって指定することができるため非常に使いやすいものとなる。

#### 【0017】

【実施例】以下本発明の実施例を図面によって詳述する。図1は本発明の一実施例を模式的に示した図で、図2は本発明によるアプリケーション連携を説明する図である。

【0018】図1において、1は、アプリケーションを管理するアプリケーションマネージャーであり、2はアプリケーション管理テーブル、3は交換データを受信側のアプリケーション、4は他の交換データを送信側のアプリケーション、5はアプリケーションがアプリケーション連携のために使用するアプリケーション連携手段、6はアプリケーション間で交換されるデータのデータ型についての処理を扱う手段であるデータユーティリティ、7は実際に交換されるデータである。

【0019】データユーティリティ6は、あらかじめ定義されているデータ型に従ってデータ型を管理しており、データ型の変換要求などに対処する。また、アプリケーションマネージャー1はアプリケーションのエンドユーザーが認識する名前と計算機システムがデータの交換を行なうために必要なアプリケーションの識別子を持つアプリケーション管理テーブルを有している。さらに、アプリケーション連携手段5においては、送信受信要求の対象データの型を設定することができ、データ型が希望の型と異なっている場合には、データユーティリティにおいて自動的に変換される。

【0020】アプリケーションを管理するアプリケーションマネージャー(1)は、アプリケーション管理テーブル(2)を内部に持ち、アプリケーション連携手段(5)による、アプリケーション管理テーブルへのアクセスを処理する。アプリケーション管理テーブル(2)は、その要素としてエンドユーザーが認識できるアプリケーションを示すアプリケーションの名前と、計算機システムがデータの交換を行なうために必要なアプリケーションの識別子からなる。たとえば、アプリケーション連携手段が、あるアプリケーション名前から計算機が識別する識別子の番号を要求した場合には、アプリケーションマネージャーが要求されている名前をアプリケーション管理テーブルの名前の欄から探し、それに対応する識別子をアプリケーション連携手段に教える。また逆に、アプリケーションの識別子からその名前を知りたいという要求に対しては、テーブルの識別子の欄を検索し該当するアプリケーション識別子を持つアプリケーションの名前を教える。

【0021】アプリケーションがアプリケーション連携のために使用するアプリケーション連携手段(5)は実際にデータを交換する際に、アプリケーションからのデータ送信受信要求を受け付け、アプリケーションマネージャーに計算機システムが認識できる識別子を受取り、該当するアプリケーションとデータ交換のためのパスをつなげ、データの送信受信を行なう。この時、要求データ型が実際のデータのデータ型と異なる場合にはデータユーティリティに対し、データのデータ型変換の依頼を行ない変換されたデータを送り先のアプリケーションに渡す。

【0022】アプリケーションは、データの送信要求、受信要求の2種類の要求ができる。まず、送信要求はあるアプリケーションが他のアプリケーションに対してデータの送信を要求する。この時、送信相手のアプリケーションは名前で指定し、相手のアプリケーションに取り込まれる時のデータのデータ型を指定することができる。また、受信要求はあるアプリケーションが他のアプリケーションに対してそのアプリケーションが所有するデータの自分に対する送信を要求する。この時、自分がデータを取り込む時のデータの型を指定することができる。

【0023】アプリケーション間で交換されるデータのデータ型についての処理を扱う手段であるデータユーティリティ(6)はデータ型の変換を行なう。これは、アプリケーション連携手段において型変更が必要となった時に、呼び出され、型変換を行なう。

【0024】交換されるデータ(7)はアプリケーション連携手段で処理されるが、この時、データのデータ型変換が必要ない場合は、送信側アプリケーション(4)から受信側アプリケーション(3)へ直接送られるが、型変換が必要な場合には送信側アプリケーションからア

アプリケーション連携手段にデータが渡された後、データユーティリティにデータの型変換が依頼され、そこで型変換されたデータが受信側アプリケーションに送られる。

【0025】次に、図2を参照しながら、あるアプリケーションが他のアプリケーションからのデータ転送を要求し、データ型の変換も含めた動作をアプリケーション連携の1つの具体例として説明する。

【0026】ここでは、名前が「アプリA」というあるアプリケーションが別のアプリケーション「アプリB」の持つデータを、「型 $\alpha$ 」というデータ型で取り込みたいものとする。ただし、「アプリB」において該当データは「型 $\beta$ 」というデータ型で存在しているものとする。

【0027】まず、「アプリA」は、アプリケーション連携手段に対して『「アプリB」からデータを「型 $\alpha$ 」という形で取り込みたい』という要求を送る。この、エンドユーザーが行うことは、アプリBからデータを取り込みたいということだけであり、相手アプリの指定もユーザーが認識できる「アプリB」と言う名前であるので、エンドユーザーにとって便利である。

【0028】この要求をアプリケーション連携手段が受け取ると、アプリケーション連携手段は、アプリケーションマネージャに対して「アプリB」という名前で登録されているアプリケーションの識別子要求する(図2(7))。

【0029】すると、アプリケーションマネージャは、アプリケーション管理テーブルを検索して、「アプリB」という名前で登録されているアプリケーションを探す。このアプリケーションがアプリケーション管理テーブルにあったならば、そのテーブルの「アプリB」に対応する欄に書いてある識別子の番号を調べ、アプリケーション連携手段に知らせる(図2(8))。

【0030】アプリケーション連携手段は識別子を受け取ったならば、その識別子に該当するアプリケーションを見つけだし、そのアプリケーションとコネクションを張る(図2(9))。

【0031】そして、「アプリB」に対し、要求されているデータの受渡しを要求し、そのデータをアプリケーション連携手段の中に取り込む(図2(10))。

【0032】アプリケーション連携手段はそのデータの型を調べる。この例では、データの型は「型 $\beta$ 」であるので、受渡しの要求されているデータ型「型 $\alpha$ 」とは異なっている。そこで、アプリケーション連携手段は、データユーティリティに対しこのデータをわたし、データの型を「型 $\beta$ 」から「型 $\alpha$ 」に変換するように要求する(図2(11))。

【0033】データユーティリティはデータを受け取り、そのデータを「型 $\beta$ 」から「型 $\alpha$ 」に変換可能な

ら、そのように変換を行い、変換されたデータをアプリケーション連携手段に返す(図2(12))。そして、そのデータを「アプリA」に渡し、データの交換を終了する。

【0034】以上のように、本実施例によればエンドユーザーは簡単な名前であつ、データ型を指定せず連携処理をすることが可能となる。

【0035】

【発明の効果】以上のように本発明は、計算機システム上で複数のアプリケーション間のデータの交換に関わるアプリケーション連携処理において、アプリケーション管理テーブルを有してアプリケーションを管理するアプリケーションマネージャと、アプリケーションがアプリケーション連携のために使用するアプリケーション連携手段と、アプリケーション間で交換されるデータのデータ型についての処理を扱う手段であるデータユーティリティとを設けることにより、アプリケーション連携手段において交換したいデータのデータ型を指定でき、交換されるデータの型が希望するデータ型と異なる場合は、データユーティリティによって自動的にデータ型の変換が行なわれ、エンドユーザーがデータ型を指定しなければいけない煩わしさから解放される。

【0036】また、アプリケーション管理テーブルを有するアプリケーションマネージャによりデータ交換希望相手のアプリケーションをエンドユーザーが認識できる名前によって指定することができるため優れたアプリケーション連携が実現できるものである。

【図面の簡単な説明】

【図1】本発明の一実施例におけるアプリケーション連携装置の模式図

【図2】本発明の一実施例におけるアプリケーション連携動作を示す概念図

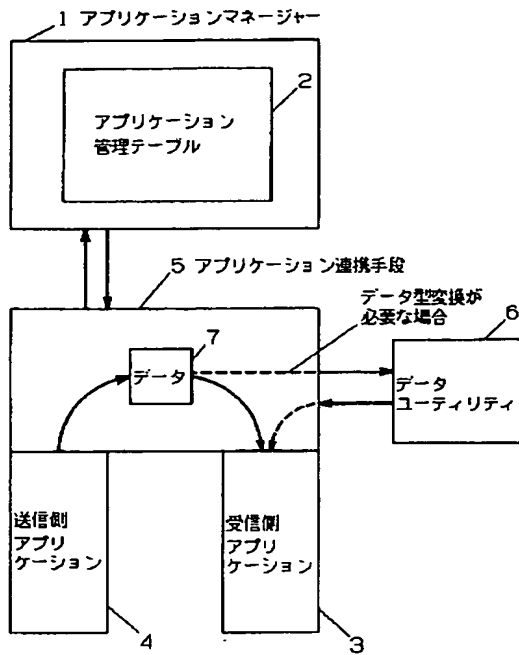
【図3】従来技術におけるカットアンドペーストの実現図

【図4】従来技術におけるドラッグアンドドロップの実現図

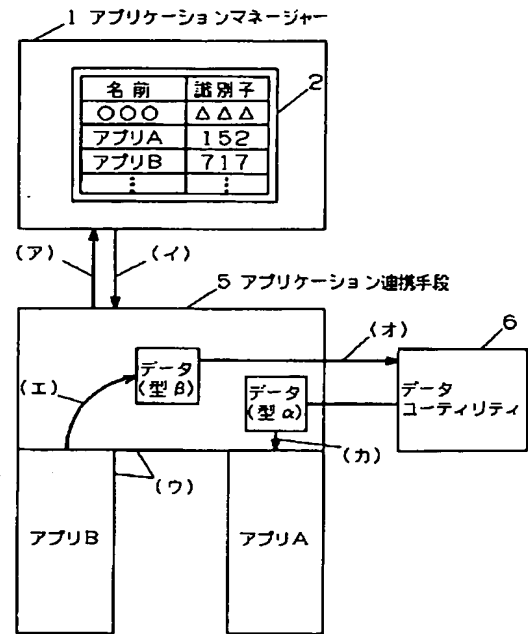
【符号の説明】

- 1 アプリケーションマネージャ
- 2 アプリケーション管理テーブル
- 3 アプリケーション
- 4 アプリケーション
- 5 アプリケーション連携手段
- 6 データユーティリティ
- 7 データ
- 31 領域
- 32 マウス移動
- 41 表示ウィンド
- 42 アプリケーション
- 43 ファイル

【図1】



【図2】



【図4】

【図3】

